

IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for scheduling program units, the method comprising:
starting a process within an operating system executing in a system having a plurality of multiple processor units, each of the multiple processor units having a plurality of processors;
starting a plurality of program units within the operating system, the program units associated with the process, wherein the program units may execute on two or more of the plurality of multiple processor units; and
upon the occurrence of a context shifting event for a first program unit of the plurality of program units, performing the tasks of:
migrating one or more the plurality of program units from one or more of the plurality of multiple processing units to one of the plurality of multiple processing units such that the plurality of program units associated with the process are executing on the same multiple processor unit,
synchronizing the scheduling of each of the plurality of program units, and
setting the context of each of the plurality of program units such that each of the program units process the same context shifting event as the first program unit.
2. (Original) The method of claim 1, wherein the program unit comprises a thread.
- 3.-4. (Canceled)
5. (Original) The method of claim 1, wherein the context shifting event comprises an exception.
6. (Original) The method of claim 5 wherein the exception comprises a signal.

7. (Original) The method of claim 1 wherein the context shifting event comprises a non-local goto.
8. (Original) The method of claim 1, wherein the context shifting event comprises a system call.
9. (Currently Amended) A system for scheduling program units, the system comprising:
~~at least one~~ a plurality of multiple processor unit units, each multiple processor unit
having a plurality of processors, wherein each of the plurality of processors on a multiple processor unit shares cache memory;
a memory coupled to the plurality of multiple processor units ~~processors~~; and
an operating environment stored in the memory and executed by at least one of the processors wherein at least one of the processors is operable to perform the tasks of:
start a process,
start a plurality of program units within an operating system, the program units associated with the process, wherein the program units may execute on two or more of the plurality of multiple processor units, and
upon the occurrence of a context shifting event for a first program unit of the plurality of program units, at least one of the processors is operable to:
migrate one or more of the plurality of program units from one or more of the plurality of multiple processing units to one of the plurality of multiple processing units such that the plurality of program units associated with the process are executing on the same multiple processor unit,
synchronize the scheduling of each of the plurality of program units, and
~~setting~~ set the context of each of the plurality of program units such that the program units process the same context shifting event as the first program unit.
10. (Original) The system of claim 9, wherein the program unit comprises a thread.

11. – 12. (Canceled)

13. (Original) The system of claim 9, wherein the context shifting event comprises an exception.

14. (Original) The system of claim 13 wherein the exception comprises a signal.

15. (Original) The system of claim 9 wherein the context shifting event comprises a non-local goto.

16. (Original) The system of claim 9, wherein the context shifting event comprises a system call.

17. (Currently Amended) A tangible computer-readable media having computer-executable instructions for performing a method for scheduling program units, the method comprising:

starting a process within an operating system executing in a system having a plurality of multiple processor units, each of the multiple processor units having a plurality of processors;

starting a plurality of program units within the operating system, the program units associated with the process, wherein the program units may execute on two or more of the plurality of multiple processor units; and

upon the occurrence of a context shifting event for a first program unit of the plurality of program units, performing the tasks of:

migrating one or more the plurality of program units from one or more of the plurality of multiple processing units to one of the plurality of multiple processing units such that the plurality of program units associated with the process are executing on the same multiple processor unit,

synchronizing the scheduling of each of the plurality of program units, and

setting the context of each of the plurality of program units such that the program units process the same context shifting event as the first program unit.

18. (Previously Presented) The tangible computer-readable media of claim 17, wherein the program unit comprises a thread.

19. – 20. (Canceled)

21. (Previously Presented) The tangible computer-readable media of claim 17, wherein the context shifting event comprises an exception.

22. (Previously Presented) The tangible computer-readable media of claim 21 wherein the exception comprises a signal.

23. (Previously Presented) The tangible computer-readable media of claim 17 wherein the context shifting event comprises a non-local goto.

24. (Previously Presented) The tangible computer-readable media of claim 17, wherein the context shifting event comprises a system call.